**INTRODUCTION**

CLOUD computing provides powerful and flexible storage services for individuals and organizations [1]. It brings about lots of benefits of sharing data with geographically dispersed data users, and significantly reduces local burden of storage management and maintenance. However, the concerns on data security and privacy are becoming one of the major obstacles impeding more widespread usage of cloud storage [2], since data owners lose physical control on their data after data are outsourced to cloud servers maintained by a cloud services provider (CSP). Data owners may worry about whether their sensitive data have been accessed by unauthorized users or malicious CSP. Cryptographic encryptions are widely suggested as standard approaches to protect the security and privacy of data outsourced to clouds [3].With encryption mechanisms, data owners first encrypt their data and then outsource to cloud servers. Then the data in clouds are stored in

ciphertext format and can only be accessed by the users having matching decryption keys. In a public cloud storage system, where different data owners may employ different encryption mechanisms according to their own data sharing requirements, it is often that a data owner wants to share his data with only one user and thus encrypts the data to generate a particular ciphertext that can only be decrypted by the specific user. However, as data sharing requirement changes, the same data owner would like to share his data with more users, which, therefore, requires to transform the ciphertext format so that multiple users can decrypt.

There are many scenarios in which the ciphertext transformation mentioned above is highly desirable. Consider group of medical insurance agents draft a health insurance plan for a client. To do so, each agent needs to collect the client’s personal information (e.g., electronic health records, occupations data, financial reports) from various data sources such as hospitals, employers, tax departments. The required data may be stored in remote cloud servers and especially, may be encrypted under different encryption mechanisms. To allow the agents to read and make use of the required data, a naive way is to let each agent acquire the corresponding decryption keys from the authorities who manage respective data. However, this would pose great concerns on data privacy. The authorities would ask a natural question: “If I give my decryption key to the agents, how to assure that all the agents would not leak the decryption key or use the decryption key to access other clients’ stored data?” This paper attempts to solve such problem technically so that the authorities can transform the ciphertexts from one encryption system to another, without handing over their decryption keys. In particular, we consider an encryption transformation mechanism that connects two types of well established encryption systems, i.e., identity-based encryption (IBE) and identity-based broadcast encryption (IBBE). We take electronic health records sharing as a motivation of our work.

Suppose a patient is equipped with implantable or wearable medical sensors to collect personal physiological records. These records are aggregated at a mobile device and then uploaded to a remote server. To protect personal privacy, the patient may encrypt his health records by some encryption mechanism, e.g., IBE, so that only his doctor can read the health records and then make proper diagnosis.

At some point, the doctor finds a complicated situation about the patient’s health and consequently, decides to consult a group of experts from different hospitals. For full understanding of the patient’s health condition, the experts first need to read the health records (see Fig. 1). Since the records are encrypted previously, the experts are impossible to directly read the data.Meanwhile, the encryption method

taken by the patient and the corresponding decryption key are unknown to the experts. This results in a dilemma for the experts: “How could we read the patient’s health records in order to provide our treatment advices?”

A trivial solution would be that the doctor first decrypts all the encrypted records and then sends out the data in plaintext (not encrypted) format to each expert. This, however, may be impractical for the doctor since a considerable computation and communication costs may be caused due to the massive health data uploaded everyday. More importantly, there is a risk of privacy disclosure by sending data in plaintext format.

There exists a cryptographic tool called proxy reencryption (PRE) that would be of help here. PRE can transform the doctor’s ciphertext into a ciphertext that can be

decrypted by one expert. Then, for n experts, PRE needs to run n times repeatedly for transferring the patient’s health data to all experts, which is inefficient. We observe that IBBE achieves a useful encryption mechanism that allows multiple users to simultaneously decrypt a ciphertext. Thus, we ask: “Can we find an efficient way to transform the encrypted data in IBE ciphertext format into an IBBE ciphertext so that multiple users can decrypt at the same time?”